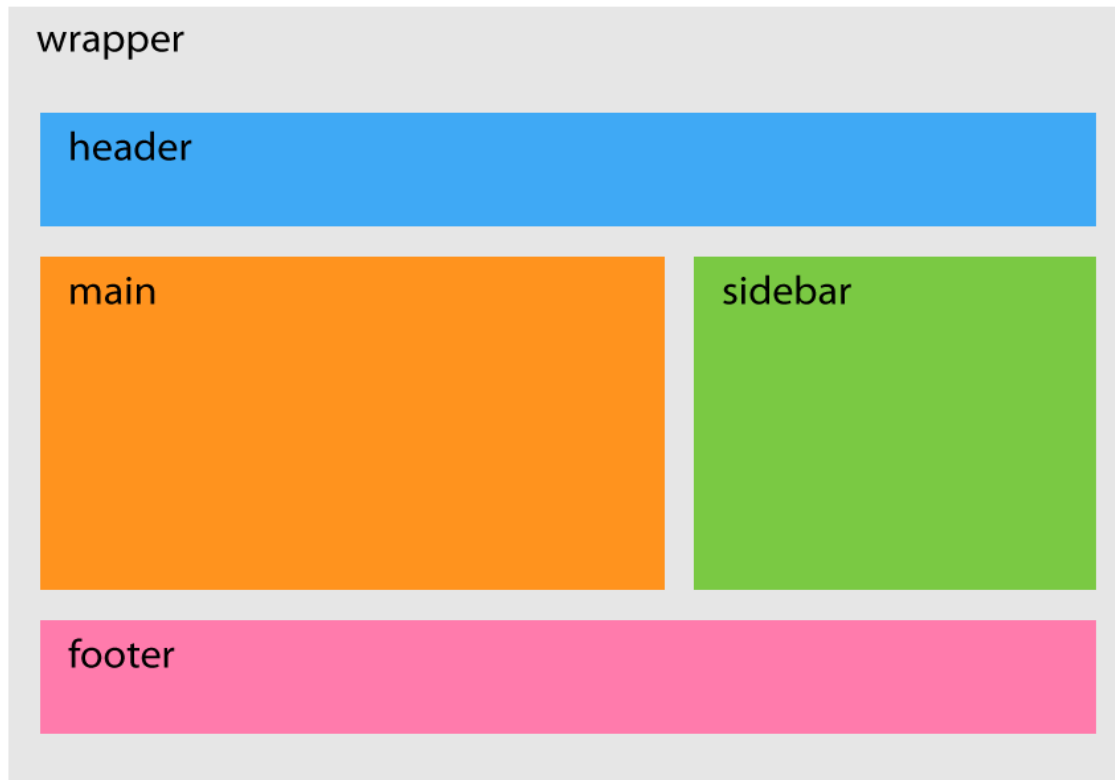


Upotreba block elemenata za layout

- Da bi kreirali osnovnu strukturu stranice, koristimo block elemente (uglavnom DIV-ove).
- Ranije su se koristile i nevidljive tabele za tu namenu, ali to danas nije preporučljivo.
- **Kreirajući DIV elemente i raspoređujući ih u celine, postavljamo osnovnu strukturu (layout) naše stranice, odnosno sajta.**

Upotreba block elemenata za layout



Float

float svojstvo sa vrednostima *left* (levo) i *right* (desno) izmešta element iz normalnog toka. Float je jedan od načina razmeštanja elemenata (tu su i *flex*, *grid* i sl.)

```
#mojElement {  
    float:left;  
    width:100px;  
}
```

Float

- Kada dodelimo ovu vrednost nekom elementu, on izlazi iz normalnog toka i ponaša se drugačije.
- Horizontalno je pozicioniran skroz levo ili desno koliko je moguće unutar svog bloka u kome je sadržan.
- Vertikalno je pozicioniran ili ispod prethodnog elementa (ukoliko taj element nije float) ili pored prethodnog elementa (ukoliko je taj element takođe float).



Na ilustraciji iznad, na delu koji je označen kao **A** vidimo više block elemenata koji su u normalnom toku (nije im postavljen float). Pod oznakom **B** vidimo jedan element kome je postavljena vrednost *float:right*; Njegova pozicija se odmah premešta na desnu stranu u visini u kojoj je originalno i bio postavljen. Svi elementi koji slede se više ne obaziru na float element i zauzimaju njegovo mesto. Ukoliko postavimo float i na sledeći element (prikaz **C**) i on će se pomeriti skroz desno do ranije float-ovanog elementa, ali se neće preklopiti. Ostali elementi se opet premeštaju ne obazirući se na ta dva.

Float

Float se postavlja veoma jednostavno, dodavanjem *float* svojstva i vrednosti *left* ili *right* u željenom CSS opisu.

```
#sidebar {  
    float:right;  
    width:200px;  
}
```

Preporuka je da float vrednost, uvek prati i širina elementa.

Float problem - naredni element

Jedna (moglo bi se reći nezgodna) osobina float svojstva je to što svaki naredni element (koji nije float) ne vidi float elemente koji mu direktno prethode, tako da će doći do problema.

Clear

clear svojstvo može imati vrednosti *left*, *right* ili *both*.

postavljajući njega na prvom narednom elementu (posle float), rešavamo problem.

```
#footer {  
    clear:both;  
}
```


Float problem 2 - roditelj element

- Još jedna nezgodna osobina float svojstva je to što **roditeljski element koji sadrži float element ne prepoznaje njegovu visinu (height).**
- Dakle, **visina float elementa ne utiče na ponašanje (visinu) roditeljskog elementa.**

Float problem 2 - roditelj element

- Roditeljski element se trudi uvek da obuhvati sve svoje elemente, osim one koji su float.
- Zašto je to bitno? Bitno je jer ako roditeljski element sadrži **samo floatovane** elemente, **neće imati visinu**, odnosno biće jednaka nuli (ukoliko nije ručno data preko height svojstva).
- Na stranici kod koje većina elemenata stoji u normalnom toku, to može dovesti do haosa.

Float, clear i overflow

- Iako zamišljen sa namerom da sređuje float, **clear** svojstvo nije uvek najsrećnije rešenje. Uglavnom se koristilo tako što se doda što jednostavniji block element direktno posle floatovanih elemenata (npr `
` tag), i koji se vizuelno sakrije jer nije potrebno da ga uopšte primetimo na strani, ali mu se i doda *clear:both*. To je možda ok rešenje, ali zahteva dodatak tog ekstra taga.
- Još jedno, možda bolje rešenje, a koje se pojavilo manje više usputno, je upotreba *overflow* svojstva. Naime, ukoliko element koji okružuje float elemente sadrži *overflow:hidden*; uzeće u obzir njihove visine, postaviće se pravilno oko njih, kao da nemaju float i tako nema potrebe za clear svojstvom. Ovom solucijom izbegavamo dodatne tagove, a i elementi koji slede ne moraju imati dodatne CSS redove.

Pitanje „zakucane“ visine

- Možda se pitate, zašto bi koristili clear ili overflow. Zašto jednostavno ne postavimo visinu (height) okružujućim elementima? Ovo bi moglo biti rešenje, ali time ograničavamo mnoge aspekte svoje stranice. Najčešće se ovi elementi koriste za osnovne elemente strukture, a to znači da će različite stranice istog sajta (sa ovim istim elementima) imati različite sadržaje.
- Postoji stara priča među web developerima koja kaže: *Ukoliko možete da duplirate sav sadržaj na stranici (dva puta duži tekst, dve slike umesto jedne i sl.) a da se layout ne raspadne, onda ste dobro postavili layout.*

Ugnježdavanje

- Prilikom planiranja treba jasno odvojiti i isplanirati elemente, odnosno definisati **koji elementi će biti na istom nivou, jedan posle drugog, a koji će se nalaziti jedan u drugom.**
- Podsećam, svi elementi se nalaze u korenorn elementu (body), ali onda možemo ostale smestati unutar njih.

Postavljanje layout-a

- Pošto smo naučili šta je float i kako možemo postaviti elemente, pogledajmo jedan opšti primer, recimo da želimo postaviti dve kolone sa centralni deo strane, main div (za sadržaj) i sidebar div (za sporedni sadržaj).

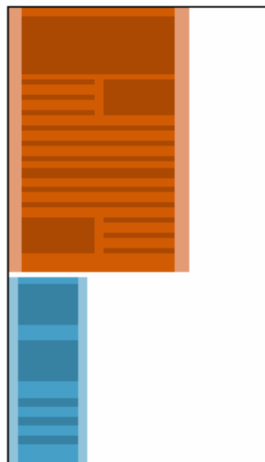
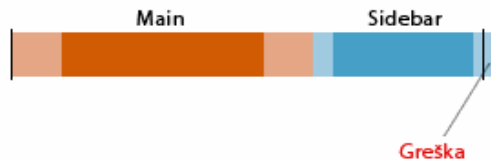
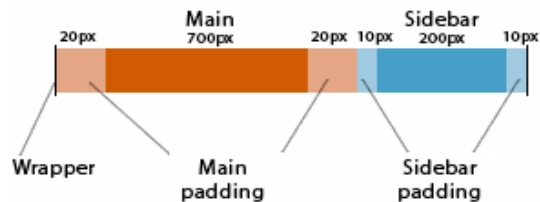
Postavljanje layout-a – HTML deo

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Float</title>
</head>
<body>
    <div id="wrapper">
        <div id="main">
            </div>
            <div id="sidebar">
                </div>
        </div>
    </body>
</html>
```

Postavljanje layout-a – CSS deo

```
#wrapper {  
    width:960px;  
}  
  
#main {  
    float:left;  
    width:700px;  
    padding:20px;  
}  
  
#sidebar {  
    float:right;  
    width:200px;  
    padding:10px;  
}
```


Ukoliko unesemo sadržaj u main i sidebar elemente, dobijamo nešto poput prikaza na sledećoj ilustraciji (levi deo slike).



Academy

Postavljanje layout-a

- Ukoliko pažljivije pogledamo levi deo ilustracije, i naš kôd, primetićemo da je ukupni zbir širina tačno 960px, što odgovara širini okružujućeg elementa.
- **700px (main) + 2x 20px (main padding) + 200px (sidebar) + 2x 10px (sidebar padding) = 960px**
- Ukoliko bi makar za 1 piksel povećali neku od ovih vrednosti, dobili bi nešto poput desnog dela ilustracije. Pošto nema dovoljno prostora (širine), sidebar bi prešao u sledeći red, odnosno ispod main dela.

Postavljanje layout-a

Naravno, ovo su samo dva elementa. Možemo sad dodati header div pre main dela (unutar wrappera) i footer posle sidebar dela (pre zatvaranja wrapper diva). Sa tom razlikom da njima nije potreban float i možemo ih pustiti da se prostiru celom širinom (po defaultu za block elemente).